Behaviour Analysis Report

Hugo Koopmans

February 21, 2013

Contents

0.1 Introduction 2
0.2 Preparing the dataset 2
0.2.1 Variabele types 2
0.2.2 Target defenition 2
0.2.3 Missing values 3
0.3 Behaviour analysis 3
0.4 Recoding 4

0.1 Introduction

This analysis report is generated using R, R-studio and knitr to sweave R code and Latex into pdf format. We have the option to include all R code that is used to generate the plots and calculations. Default this feauture is dissabled.
The behaviour analysis step is the second step in a datamining analysis.
Steps identified in the datamining process are: - Data analysis - Behaviour analysis - Missing value analysis - Missing value imputation (optional) - Binning - Feature selection - Model development - Model analysis - Model deployment

0.2 Preparing the dataset

Basic information on dataset:

```
# goto data file directory
setwd("~/r-studio/TdH")

#### parameters #### filename
filename <- "data-analysis.tab"
# sample size in percentage
ssize = 10

# read dataframe from tab delimets file
data <- read.delim(filename)

# determine number of rows and colums in dataframe
rows <- nrow(data)
colums <- ncol(data)
```

We are using data from file : data-analysis.tab.
The dataset has 34 variables and 59992 rows.

0.2.1 Variabele types

```
# datasetstructure <- str(data)
var_names <- names(data)
num_var_names <- names(data[sapply(data, is.numeric)])
num_vars <- length(num_var_names)
cat_var_names <- names(data[sapply(data, is.factor)])
cat_vars <- length(cat_var_names)
# target definition
target_name <- "nudona"
```

The following variabele are present in the dataset:
jaarbedr, minbegjr, proj, nudona, r20102009mndbedrag, r20112010mndbedrag, X2009avgbedragmnd, X2010avgbedragmnd, X2011avgbedragmnd, bronaktie, bronlaatstebetwijze, Jaarbinnen, sexe, leeftijd2011, Bronbinnen, mailingen, magpost, magdigi, diginws, TM, mail07, catHHINKOMEN, catHHSOCIALE, catHHOPLEIDI, catHHLEVENSF, catHHGEOTYPE, catHHTYPEWO, catHHEIGENDO, catHHWOZWAA, catBELEGGERS, catLENERS, catSPAARDERS
catSWITCHGEVO, catMERKENTROU
We have 26 numeric variables and 8 categorical variables (or factors in R).

0.2.2 Target defenition

This analysis aims to report of the behaviour of each individual 'predictor' to a target variable. The target variable should be a categorical variable having two categories(or factor levels).

The target variable is nudona.
The target hass the following proportion of outcomes:

```
# check if mising values in target
if (length(which(is.na(data[[target_name]]))) > 0) {
    cat("Warning : Removing", length(which(is.na(data[[target_name]]))), "cases with missing target val
    data <- subset(data, !is.na(data[[target_name]]))
}

# check if target is a factor if not make it a factor
if (is.numeric(data[[target_name]])) {
    data$target <- as.factor(data[[target_name]])
} else {
    data$target <- data[[target_name]]
}

# display counts and percentage on target
library(xtable)
t <- cbind(table(data$target), 100 * prop.table(table(data$target)))
xt <- xtable(t)
digits(xt) <- c(0, 0, 2)
names(xt) <- c("count", "%")
print(xt)
```

	count	%
0	6319	10.53
1	53673	89.47

0.2.3 Missing values

Before we can do model analysis we need to take care of missing values. The simplest appraoch is to delete cases including one or more missing entries but this can remove a large proportion of valuable data.
We can also remove individual variables if they have a high percentage of missing atributes.
Or we can replace or impute missing data with for instance an average or most frequent value. Actually changing the data. For this we use kNN nearest neighbors.

For now we will throw away all cases that have one or more missing attribute.

This dataset has 31094 complete cases out of 59992, which is 0.5183 percent.

```
# handel cases with missing data if number of cases that have missing data
# is limited the just drop the missing otherwise we need more advanced
# replacement or imputation mechanisms or drop the variable that has the
# misssings

# list rows of data that have missing values
# length(which(!complete.cases(data)))

# create new dataset without missing data
data <- na.omit(data)

# kNN missing value imputation todo get this to work in parrallel
# cleandata <- knnImputation2(data,k=5)
```

0.3 Behaviour analysis

The selected inputs have the following raw predictive capacity:

```
# kendall tau over all predictors

# method still slow , sampling needed
```

```
n <- round(nrow(data) * ssize/100)
s_data <- data[sample(nrow(data), size = n), ]

# create indices for all colums we want to calc correlation measure
drops <- c("id", "target")
idx <- which(!(names(data) %in% drops))

# function to calc correlation measure
myCorMeasures <- function(i = 1, target = "target", df = data) {
    result <- cor.test(xtfrm(df[, i]), xtfrm(df[, target]), alternative = "two.sided",
        method = "kendall")$estimate
    return(result)
}

# todo add gini and other quality measures

# list of correlations of variables to target
correlation.Tau <- lapply(idx, myCorMeasures, df = s_data)

# # begin parrallel stuf library(parallel)
#
# # set up the cluster size_of_pool = 4 cl <- makeCluster(size_of_pool) #
# do things in parrallel correlation.Tau <- parLapply(inds,myCorMeasures)
#
# stopCluster(cl) # end parrallel stuff

# make a numeric vector
nct <- as.numeric(correlation.Tau)
# wrld_data[order(wrld_data$NAME),] dd[with(dd, order(-z, b)), ]
library(xtable)
name <- names(data[, inds])

## Error:  object 'inds' not found

t <- data.frame(name, nct)

## Error:  object 'name' not found

# sort correlation ascending
t_sorted <- t[with(t, order(nct)), ]

## Error:  numeric 'envir' arg not of length one


xt <- xtable(t_sorted)

## Error:  object 't_sorted' not found

digits(xt) <- c(0, 2, 4)
names(xt) <- c("variable", "Kendall Tau correlation")
print(xt, table.placement = "H")
```

	variable	Kendall Tau correlation
0	6319.00	10.5331
1	53673.00	89.4669

0.4 Recoding

Next step is model building.